Reliable Gold Code Generators for GPS Receivers

Mohamed Mourad Hafidhi and Emmanuel Boutillon
Lab-STICC, UMR 6582, Université de Bretagne Sud
56100 Lorient, France
email: mohamed.hafidhi@univ-ubs.fr, emmanuel.boutillon@univ-ubs.fr

Chris Winstead
ECE Dept., UMC 4120, Utah State University
Logan, UT 84322, USA
email: chris.winstead@usu.edu

***Abstract*—This paper[1] examines five strategies for upset protection in Gold sequence generators used to maintain signal acquisition in GPS receivers. This work is motivated by the increased prevalence of single-event upsets in ultra low-power GPS receivers. If any upset occurs in the Linear Feedback Shift Register (LFSR) modules, then the corresponding satellite signal must be re-acquired, resulting in high energy expenditure and time delay. We evaluate the performance and complexity of methods based on error correction and modular redundancy.**

I. Introduction

Global Positioning System (GPS) receivers are heavily used in mobile contexts, and there is motivation to minimize power consumption and maximize battery life in these devices. When operating at minimal power levels, there is greater occurrence of internal logic upsets due to noise processes and external interference in combination with process/voltage/temperature (PVT) variation [1], and these problems increase with the advancement of CMOS technology [2]. Logic upsets can manifest as momentary faults in the device's behavior, or as persistent faults that require re-start or signal reacquisition. Fault tolerance has been studied for data fusion in navigational systems that incorporate GPS receivers along with other sensor devices. For example, the authors of [3] consider an intelligent data fusion system to compensate for delayed or erroneous GPS data due to acquisition loss or signal obstruction. To our knowledge, there has not been a detailed study of options for internal fault-tolerance to prevent acquisition loss from occuring in GPS receivers.

GPS systems use a family of ranging signals called Course/Acquisition (C/A) codes belonging to the family of Gold pseudo random noise (PRN) sequences. The C/A codes are generated from the product of two 1,023-bit PRN sequences called G1 and G2. Both G1 and G2 are generated by 10-stage LFSR modules [4]. Fig. 1 shows the standard G1 generator. If the LFSR register states are disturbed, then the corresponding C/A code state becomes corrupted, resulting in loss of the GPS signal tracking. This problem can be mitigated if the LFSR state registers are protected by an error correction method. This paper presents five solutions to make a system formed of four G1 registers (four is the minimum number of satellites to determine the position of a receiver) more robust to upset errors, and evaluates those solutions in terms of their performance benefit and gate overhead. These solutions can be extended and generalized for any type of LFSR register. Values stored in all registers, at the clock cycle t, will be represented by the matrix $R^{(t)} = (r_{i,j}^{(t)})$ with $1 \leq i \leq 4$ and $1 \leq j \leq 10$, where j is the j^{th} position of the i^{th} LFSR.

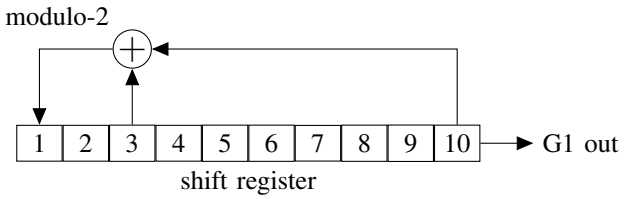


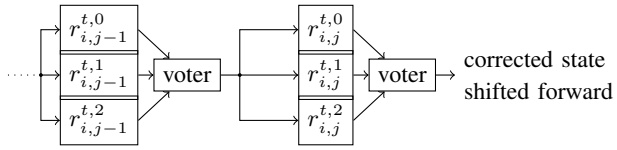
Fig. 1. G1 sequence generator.

The remainder of this paper is organized as follows. Sec. II describes two Triple Modular Redundancy (TMR) methods, Sec. III presents a row×column parity-check solution, and Sec. IV presents two solutions based on Hamming codes. Each of these sections evaluates the solutions' performance by computing the mean time to failure (MTTF). Sec. V provides synthesis and performance results, and compares the complexity as the number of equivalent NAND gates for each solution. Finally Sec. VI offers conclusions.

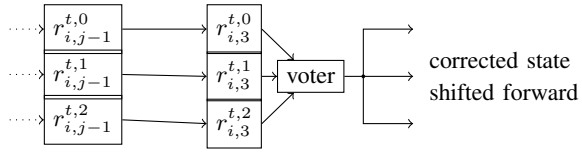
II. Triple Modular Redundancy (TMR)

TMR is a classical solution for fault tolerance in electronic systems [5]. In a TMR system, the original module (i.e. the LFSR) is replicated three times, and error correction is achieved by a majority vote operation. In our analysis, we assume every flip-flop is triplicated and assigned a single voter as shown in Fig. 2(a). We note that there are several alternative configurations, such as the triple-voter method or the restorative-feedback voter [6], but to minimize overhead we only consider the most basic TMR strategy. If an upset occurs in any one of the three LFSR modules, the other two devices can correct and mask the fault. But once two of the three devices fail an uncorrectable failure results. An uncorrectable failure is inevitable in a long-running LFSR, so we evalute the system's reliability by calculating the system's Mean Time to Failure (MTTF): larger MTTF implies a more reliable system.

[1]This work has received a French government support granted to the COMIN Labs excellence laboratory and managed by the National Research Agency in the "Investing for the Future" program under reference ANR-10-LABX-07-01. It has also received support from the Brittany Region. Support was also from the US NSF award ECCS-0954747 and the Franco-American Fulbright Commission.



(a) Multi-column TMR



(b) One-column TMR

Fig. 2. Shift register with TMR protection in each flip-flop, where $R_{i,j}^{t,x}$ denotes the x^{th} replica of the j^{th} flip-flop in the i^{th} LFSR of a G1 sequence generator group at the step t.

To compute the MTTF it is first necessary to solve the probability distribution of system failures. It is assumed that upset events in the replicated flip-flops are independent. Let p be the flip-flop upset probability per time step (i.e clock cycle), and $\mathbf{X_{k,t}}$ a random variable describing the upset state of the k^{th} bit in an LFSR at time step t. The stored value is either correct ($\mathbf{X_{k,t}} = 0$) or erroneous ($\mathbf{X_{k,t}} = 1$). Given no overall failure prior to time step $t-1$, $\Pr(\mathbf{X_{k,t-1}} = 1) = p$ for all k. After applying the majority vote operation, the probability of a correct state at the voter's output, q_v, is given by:

$$q_v = (1-p)^3 + 3p\,(1-p)^2\,. \tag{1}$$

Now let p_c be the probability that the entire LFSR state is correct after applying majority vote operations on all flip-flops:

$$p_c = \prod_{k=1}^{10} q_v$$

Let P_{Fail} be the probability of an instantaneous failure in the complete system. This probability is expressed as

$$P_{\text{Fail}} = 1 - {p_c}^4. \tag{2}$$

Finally, we calculate $P_e(t)$, the probability that the failure of the complete system occurs at time step t:

$$P_e(t) = (1 - P_{\text{Fail}})^{(t-1)} P_{\text{Fail}}. \tag{3}$$

Since $P_e(t)$ has a geometric distribution, the mean of this distribution is well known and yields the MTTF:

$$\text{MTTF} \triangleq \sum_{t=1}^{\infty} t P_e(t) = \frac{1}{P_{\text{Fail}}}. \tag{4}$$

As an example, given a flip-flop upset probability $p = 10^{-3}$, we find $P_{\text{Fail}} = 1.2 \times 10^{-4}$, and the mean time to failure (MTTF) is equal to 8339 cycles.

In the TMR solution the vote operation is done for every bit of every LFSR. To reduce the complexity, it is possible to perform voting in a single column as described in Fig. 2(b). Since the data is shifted in a circular pattern around the LFSR,

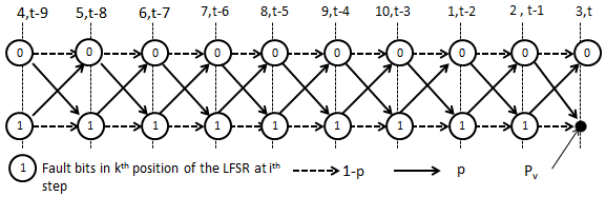


Fig. 3. Trellis graph describing error propagation in the one-column error correction solution.

any single error will eventually pass through the voter where it can be corrected. The LFSR has length ten, so a single error may persist in the system for at most ten clock cycles. One-column voting tends to decrease (i.e. worsen) the MTTF of TMR. In order to model the propagation of errors through the LFSR, we use the trellis model shown in Fig. 3, which assumes the voter is placed in column three. By choosing this position of the voter, we garantee that every error in this system will pass by the voter and will be corrected in at most 10 clock cycles. Given that no system failure has occured prior to time step t, we only need to consider a trellis depth of ten stages, since a successul correction would have eliminated any errors in the voter's column at time $t-10$. Let $p_f(t-\tau)$ be the error probability in a flip-flop τ time-steps before it's data reaches the voter. Clearly $p\,(t-10) = 0$, and for subsequent stages we have

$$\begin{aligned} p_f\,(t-\tau) &= p\,(1 - p_f\,(t-\tau-1)) \\ &\quad + (1-p) p_f\,(t-\tau-1)\,. \end{aligned} \tag{5}$$

Iterating this calculation in the trellis model, we determine p_v, the probability of flip-flop error in the voting column, as the second element in the state vector P_v given by

$$P_v = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}^{10} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{6}$$

Using this model, we find the probability of a correct state at the one-column voter's output, q_v, by substituting p_v in place of p in (1). Then the instantaneous failure probability for the four-LFSR system is

$$P_{\text{Fail}} = 1 - q_v^4. \tag{7}$$

The MTTF, in this case and for $p = 10^{-3}$, is equal to 854 clock cycles, nearly ten times lower than TMR. The one-column approach trades reliabiliy for complexity gains, as evaluated and compared in Sec. V.

III. Method with Parity-Check Error Detection

In this section, we will modify the architecture of each LFSR by adding a parity flip-flop to each row. We define the row parity $v_i^{(t)}$ as the modulo-2 sum over all ten values in a single LFSR i at time step t, hence $v_i^{(t)} = \bigoplus_{j=1}^{10} r_{i,j}^{(t)}$.

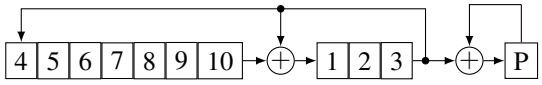


Fig. 4. New architecture of G1 LFSR with row parity.

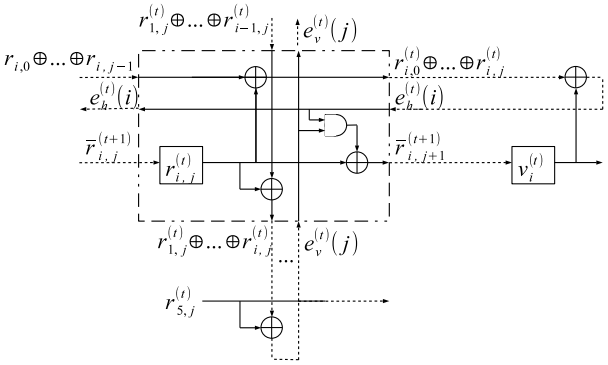


Fig. 5. Schematic for parity-based correction of the j^{th} flip-flop in the i^{th} LFSR.

Given $r_{i,1}^{(t+1)} = r_{i,3}^{(t)} \oplus r_{i,10}^{(t)}$, the expression for $v_i^{(t+1)}$ can be simplified as

$$v_i^{(t+1)} = \bigoplus_{j=1}^{10} r_{i,j}^{(t+1)} = (r_{i,3}^{(t)} \oplus r_{i,10}^{(t)}) \oplus (\bigoplus_{j=1}^{9} r_{i,j}^{(t)})$$
$$= r_{i,3}^{(t)} \oplus (\bigoplus_{j=1}^{10} r_{i,j}^{(t)}) = v_i^t \oplus r_{i,3}^{(t)}. \quad (8)$$

Based on this equation, we add the new parity flip-flop to every LFSR in the system, according to the architecture shown in Fig. 4. In order to detect errors in LFSR i, we independently compute the error as $e_h^{(t)}(i) = (\bigoplus_{j=1}^{10} r_{i,j}^{(t)}) \oplus v_i^{(t)}$. If $e_h^{(t)}(i) \neq 0$, an odd number of errors has occurred.

The row-parity solution based on Fig. 4 is sufficient to detect errors but cannot identify their position. In order to detect the location of errors, a new LFSR is added to the system comprised of the column-parity, so that

$$r_{5,j}^{(t)} = \bigoplus_{i=1}^{4} r_{i,j}^{(t)}. \quad (9)$$

Since the parity LFSR is identical to the others, it only needs to be initialized with (9) at a single time t_0. Under error-free operation, the condition described by (9) is preserved in the LFSR state evolution. The new LFSR sytem consists of five LFSRs: The fifth row is the parity LFSR, and the eleventh column is the set of parity-check bits corresponding to Fig. 4. The column and row errors are detected as $e_v^{(t)}(j) = (\bigoplus_{i=1}^{4} r_{i,j}^{(t)}) \oplus r_{5,j}^{(t)}$ and $e_h^{(t)}(i)$, respectively. If there is a row l where $e_h^{(t)}(l) \neq 0$, then we should find all positions q for which $e_v^{(t)}(q) \neq 0$. Errors are corrected by flipping the bit in these positions q using the XOR function as shown in the architecture in Fig. 5. Corrected register values are denoted as $\bar{r}$ in this figure. With this method, an odd number of errors can be corrected if they occur in only one row. The probability of a system failure in this case is

$$P_{\text{Fail}} = 1 - \left((1-p)^{55} + 55p(1-p)^{54} \right). \quad (10)$$

Continuing the example from the previous sections with $p = 10^{-3}$, we find that $P_{\text{Fail}} = 0.0014$ and the MTTF is 697. This is far below the performance of TMR. Due to its simplicity, the parity method provides good complexity gain as discussed in Sec. V.

IV. Solution using Hamming codes

Researchers previously considered the theoretical characteristics of multi-LFSR systems protected by linear error correcting codes. For example, Hadjicostis and Verghese considered the asymptotic properties of a large group of identical LFSRs jointly encoded using LDPC codes [7]. In this section, we describe a similar but much smaller scale approach using a basic Hamming code. As in the row/column parity solution, we are using in this method parity LFSRs, which are constructed as linear transformations of the original system. In order to jointly encode the consituent LFSRs and produce valid parity LFSRs, the system must obey to two criteria. First they should be described by the same polynomial as all LFSRs in the original system. Second, the initial states of the parity LFSRs, represented by the matrix $R^{(0)}$, are obtained using a Hamming encoding operation:

$$C^{(0)} = G^T \times R^{(0)}$$

where G^T is the transpose of the generator matrix G of the (7,4) Hamming code and $R^{(0)}$ represents initial states of the original four LFSRs. For this encoding we use the standard Hamming generator matrix given by

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Since the LFSR states are updated by a linear transformation at each time step, if no errors are generated in the system, then the parity LFSRs will preseve the Hamming code structure [7], i.e. at any time step t,

$$C^{(t)} = G^T \times R^{(t)}.$$

Let $Q^{(t)}$ be the new encoded system, defined by $\mathrm{Q}^{(\mathrm{t})} \triangleq \begin{pmatrix} S^{(t)} \\ C^{(t)} \end{pmatrix}$. The system is now a matrix with 10 columns $\mathrm{Q}^{(\mathrm{t})} = (Q_1^{(t)}, \cdots, R_{10}^{(t)})$. By decoding all columns of $Q^{(t)}$, we are able to detect and correct errors present in each column decoded at a step t. We assume a syndrome decoding method, which is able to correct a single error. The system in this case fails once two errors are generated in the same column. The probability that a failure is not declared in one column is:

$$q_h = (1-p)^7 + 7p\,(1-p)^6. \quad (11)$$

TABLE I
SYNTHESIS RESULTS FOR EACH METHOD, SHOWING TOTAL MODULE COUNTS AND THE EQUIVALENT NUMBER OF NAND GATES PER LFSR

Method	FF	XOR2	AND2	AND3	Maj	NAND eq.
TMR	120	12	0	0	40	242.5
TMR (one col.)	120	12	0	0	4	220
Hamming	70	167	0	70	0	235.375
Hamming (one col)	70	23	0	7	0	139.5
Parity-Check	55	159	55	0	0	209.375

The probability of an instantaneous failure in the system is:

$$P_{\text{Fail}} = 1 - \prod_{k=1}^{10} q_h \tag{12}$$

Using this method, with $p = 10^{-3}$, the MTTF is equal to 4713 clock cycles. To reduce the complexity, decoding can be performed only in one column. To determine the MTTF for this method, we modify the method used for the one-column TMR method. Since Hamming decoding is performed for a single column only, the flip-flop error propagation is modeled by the trellis process in Fig. 3, and the probability of flip-flop error in the decoder's column is p_v as given by (6). Since a system failure is produced by two or more errors among the flip-flops in the decoded column, we find that

$$P_{\text{Fail}} = 1 - \left[(1-p_v)^7 + 7p_v\,(1-p_v)^6\right]. \tag{13}$$

We find, with $p = 10^{-3}$, MTTF $= 501$. Once again the MTTF is made worse, with a possible gain in complexity.

V. Complexity and Performance Comparison

All of the described methods were evaluated in term of hardware complexity and MTTF. The designs are composed of simple components: flip-flops (FF), 1bit XOR2 modules, and 3-input AND (AND3) gates. The associated equivalent NAND complexity is evaluated for each synthesis product as reported in Table I. Given a probability of an upset in one flip flop, $p = 10^{-3}$, results from the analyse of the mean time to failure and the complexity of diferents method are summarised in Fig. 6. From this figure we can see that the full TMR method provides MTTF close to ten thousand clock cycles, but this is more than what is needed since the PRN are of length 1023. The full Hamming method's MTTF of 4713 is not really a loss since all registers are reinitialized to ones every 1023 cycles. Now by varying the probability of the upset p, the MTTF of all solutions are compared in Fig. 7. The 1023 MTTF threshold is shown as a dashed line in Figs. 6 and 7.

VI. Conclusion

Our results show a roughly exponential improvement in MTTF with increasing gate complexity. The best choice of protection depends on the technology's upset probability. For a high upset probability, the full Hamming or TMR method may be necessary to achieve an acceptable MTTF. For a small upset probability, the the Hamming (one col.) method provides the required MTTF, but reduces the gate complexity by 42% compared to full TMR.

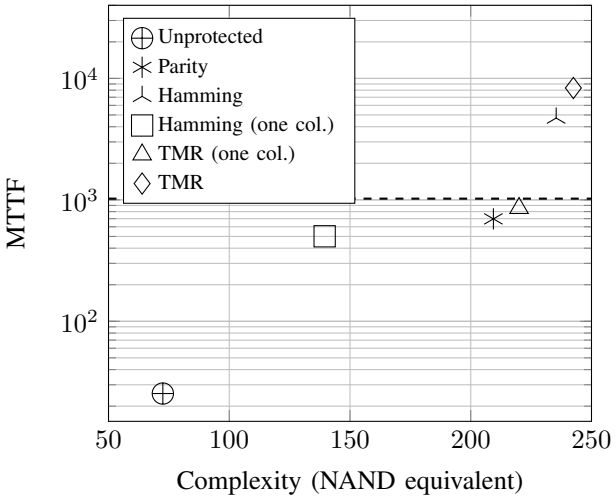


Fig. 6. Complexity and performance of the proposed methods with $p = 10^{-3}$. The dashed line indicates the PRN length.

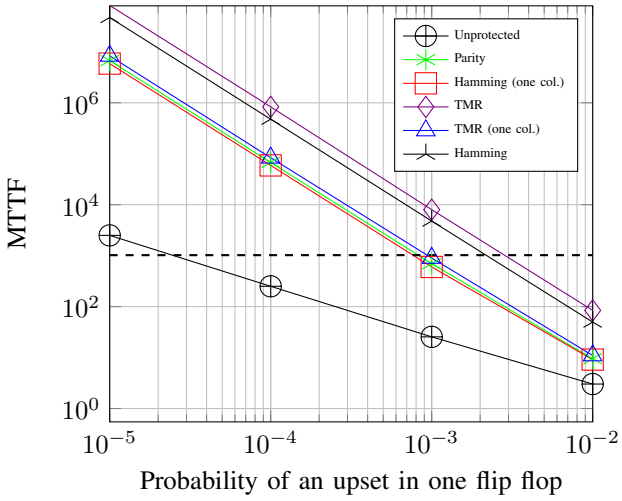


Fig. 7. Performance as a function of the upset probability p. The dashed line indicates the PRN length.

References

[1] S. Borkar, "Tackling variability and reliability challenges," *IEEE Design and Test of Computers*, vol. 23, no. 6, 2006.

[2] P. B. J. A. Jayanth Srinivasan, Sarita V. Adve, "The impact of technology scaling on lifetime reliability," *The International Conference on Dependable Systems and Networks (DSN-04)*, 2004.

[3] M. Jaradat and M. Abdel-Hafez, "Enhanced, delay dependent, intelligent fusion for ins/gps navigation system," *Sensors Journal, IEEE*, vol. 14, no. 5, pp. 1545–1554, May 2014.

[4] J. B.-Y. Tsui, *Fundamentals of Global Positioning System Receivers*. Wiley-Interscience, 2000.

[5] R. E. Lyons and W. Vanderkulk, "The use of triple-modular redundancy to improve computer reliability," *IBM J. Res. Dev.*, vol. 6, pp. 200–209, 1962.

[6] C. Winstead, A. Tejeda, E. Monzon, and Y. Luo, "Error correction via restorative feedback in M-ary logic circuits," *J. of Mult. Valued Logic and Soft Comp.*, vol. 23, no. 3–4, 2014.

[7] C. Hadjicostis and G. C. Verghese, "Coding approaches to fault tolerance in linear dynamic systems," *Information Theory, IEEE Transactions on*, vol. 51, no. 1, pp. 210–228, 2005.